

Programación I
Grado de Ingeniería Informática y Doble Grado II-ADE - ETSIINF - UPM
Examen Eval1
30/10/2018

Realización: El ejercicio se realizará en la hoja de respuestas, que será lo único que se entregará. En ella se harán constar los apellidos y el nombre. Se pueden utilizar hojas de sucio aparte.

Duración: El tiempo para realizar este examen es de **1 hora**.

Calificaciones: Las calificaciones se publicarán el **5 de Noviembre**.

Revisión: Se mandará un correo con la información relativa a la revisión del examen.

Enunciado

Ejercicio 1

En español hay cinco vocales: a, e, i, o, u. Podemos decir que la **a** ocupa la posición **1** dentro de las vocales, la **e** la posición **2** y así hasta decir que la **u** ocupa la posición **5**.

Se pide:

1.1 (1 punto) Implementar en Java la función **posVocal** que dada una vocal **v** devuelva un número entero que indique su posición según lo explicado. Como precondition **v** sólo puede ser una vocal en minúscula.

1.2. (1 punto) Escriba dos pruebas que sirvan para comprobar que, al llamar a la función **posVocal** con dos valores distintos, el resultado es el esperado.

1.3 (1 punto) Escriba una función **sumPosVocales** que, dadas dos vocales **v1** y **v2**, devuelva la suma de las posiciones que ocupan dichas vocales. Por ejemplo, si las vocales son **a** (posición 1) y **e** (posición 2), el resultado es **3** y si las vocales son **a** (posición 1) y **o** (posición 4), el resultado es **5**. Como precondition **v1** y **v2** sólo pueden ser vocales en minúscula.

Ejercicio 2

Dado el siguiente código:

```
static String queHago(int a, int b){
    String resultado = "";
    for (int i = a; i <= b; i = i + 1){
        resultado = i + " " + resultado;
    } // de for
    return resultado;
} // de queHago
```

Se pide:

2.1 (1 punto) ¿Cuál sería el resultado de la llamada **queHago(2,5)**?

2.2 (1 punto) ¿Cuál sería el resultado de la llamada **queHago(8,8)**?

Nota: Las operaciones en sucio, traza o evolución de variables en memoria que necesites para llegar al resultado debes incluirlas en la hoja de respuestas. Si no se incluyen estas operaciones en sucio las respuestas al ejercicio no se considerarán válidas.

Ejercicio 3

Dado el siguiente código:

```
//PRE: f >= 0
static int enigma(int f){
    if (f < 10)
        return f;
    else
        return f % 10 + enigma(f / 10);
} // de enigma
```

3.1 (1 punto) ¿Cuál sería el resultado de la llamada **enigma(123)**?

3.2 (1 punto) ¿Cuál sería el resultado de la llamada **enigma(46)**?

Nota: Las operaciones en sucio, traza o evolución de variables en memoria que necesites para llegar al resultado debes incluirlas en la hoja de respuestas. Si no se incluyen estas operaciones en sucio las respuestas al ejercicio no se considerarán válidas.

Ejercicio 4 (3 puntos)

Una progresión aritmética es una sucesión de números en la que el elemento siguiente a uno dado se calcula sumando una constante denominada diferencia. Por ejemplo, si la diferencia fuera 3, el valor siguiente a 1 sería 4 y el valor siguiente a 4 sería 7. Se pide escribir en Java una función que devuelve el número de elementos de una progresión aritmética. La cabecera de dicho método es:

```
static int elemProgAritmetica (int primero, int diferencia, int ultimo)
```

donde **primero** es el primer número de la progresión, **diferencia** es la constante y **ultimo** es el ultimo valor de la progresión. Estos tres números son siempre mayores que **0**.

Ejemplos:

`elemProgAritmetica(1,3,7) → 2`

`elemProgAritmetica(4,2,10) → 4`

Examen Eval1 - 30/10/2018 - Hoja De Respuestas

Apellidos: _____	Nombre: _____
DNI: _____	Grado(II+ADE): _____

Ejercicio 1

1.1

```
static int posVocal (char c){  
    switch (c){  
        case 'a': return 1;  
        case 'e': return 2;  
        case 'i': return 3;  
        case 'o': return 4;  
        default: return 5;  
    } // de switch  
} // de posVocal
```

1.2

```
System.out.println("posVocal('a') = " + posVocal('a'));  
System.out.println("posVocal('o') = " + posVocal('o'));
```

1.3

```
static int sumPosVocales (char c1, char c2){  
    return posVocal(c1) + posVocal(c2);  
} //
```

Ejercicio 2

2.1 queHago(2,5), a = 2, b = 5

Inicialmente resultado = ""

Paso 1: i = 2, 2 <= 5? TRUE entonces resultado = 2 + " " + "" = "2 "

Paso 2: i = 3, 3 <= 5? TRUE entonces resultado = 3 + " " + "2 " = "3 2 "

Paso 3: i = 4, 4 <= 5? TRUE entonces resultado = 4 + " " + "3 2 " = "4 3 2 "

Paso 4: i = 5, 5 <= 5? TRUE entonces resultado = 5 + " " + "4 3 2 " = "5 4 3 2 "

Paso 5: i = 5, 6 <= 5? FALSE

termina de ejecutarse la función y el resultado es "5 4 3 2"

2.2 queHago(8,8), a = 8, b = 8

Inicialmente resultado = ""

Paso 1: i = 8, 8 <= 8? TRUE entonces resultado = 8 + " " + "" = "8 "

Paso 2: i = 9, 9 <= 9? FALSE

termina de ejecutarse la función y el resultado es "8 "

Ejercicio 3

3.1 `enigma(123)`, `f = 123`

`123 < 10? FALSE → 123 % 10 + enigma(123 / 10) → 3 + enigma(12)`

`enigma(12) → 12 < 10? FALSE → 12 % 10 + enigma(12 / 10) → 2 + enigma(1)`

`enigma(1) → 1 < 10? TRUE → 1`

`enigma(123) = 3 + 2 + 1 = 6`

3.2 `enigma(46)`, `f = 46`

`46 < 10? FALSE → 46 % 10 + enigma(46 / 10) → 6 + enigma(4)`

`enigma (4) → 4 < 10? TRUE → 4`

`enigma(46) → 6 + 4 = 10`

Ejercicio 4

```
static int elemProgAritmetica(int primero, int diferencia, int ultimo){  
    int resultado = 0;  
    for (int i = primero; i <= ultimo; i = i + diferencia){  
        resultado++;  
    } // de for  
    return resultado;  
} // de elemProgAritmetica
```